3D Model Based Pose Estimation For Omnidirectional Stereovision

Guillaume Caron, Eric Marchand, E. Mouaddib

3D Model Based Pose Estimation For Omnidirectional Stereovision

Guillaume Caron, Eric Marchand and El Mustapha Mouaddib

Abstract— **Robot vision has a lot to win as well with wide field of view induced by catadioptric cameras as with redundancy brought by stereovision. Merging these two characteristics in a single sensor is obtained by combining a single camera and multiple mirrors. This paper proposes a 3D model tracking algorithm that allows a robust tracking of 3D objects using stereo catadioptric images given by this sensor. The presented work relies on an adapted virtual visual servoing approach, a non-linear pose computation technique. The model take into account central projection and multiple mirrors. Results show robustness in illumination changes, mistracking and even higher robustness with four mirrors than with two.**

I. INTRODUCTION

Robot localization is a complex task and vision has shown its interests along last decades. Whatever the kind of camera is, several works have been done to do self-localization using vision. The choice of the sensor is critical for the aimed application. Each of them has pros and cons but some, such as omnidirectional sensors, have really interesting advantages. Their major interest is to allow to observe a landmark during a long period of time, all around the robot, which is synonym of precision. Nevertheless, the problem of depth estimation is still present.

Stereovision sensors have the interesting property of allowing estimation of the depth. They are, for instance, composed by two perspective cameras, as human vision system. Knowing the geometry of the sensor, one can recover a point depth.

So an interesting idea is to use a sensor that merges stereovision and omnidirectional vision. This is one of these sensors (a small review of them can be found in [1]) we used for the work presented in this paper. The sensor was designed by Mouaddib *et al.* in [2]. It is a catadioptric sensor composed by a unique camera and four parabolic mirrors placed in a square at the same distance from the camera (Fig. 1).

Our aim is to do self-localization using this sensor since its potential in terms of robustness and precision is high. We propose to investigate the visual servoing approach to do this task.

Visual servoing aims to move a camera usually mounted on the robot end-effector in order to reach a desired pose using image information. For our problem, one can imagine to virtualize the camera, starting from an initial pose and moving to reach the real pose of the camera, still using image information. This is the Virtual Visual Servoing (VVS) framework introduced by Sundareswaran *et al.* in [4] and then by Marchand *et al.* in [5], a full scale non-linear optimisation technique which can be used for pose computation. Some works have been done, for perspective camera [6], stereo perspective rig [7] and even for monocular omnidirectional vision [8]. These works also deal with corrupted data, which is frequent in image feature extraction, using the widely accepted statistical techniques of robust M-estimation [9]. The goal in this paper is to develop robust VVS for omnidirectional stereovision to recover the camera pose and to show qualities and contributions of the presented sensor for this kind of application.

Fig. 1. Our sensor: orthographic camera, parabolic mirrors, 90 cm height

We first present sensor modelling before developping the pose estimation approach. After that, the chosen feature type will be tackled as well as image tracking used method. Finally, experimental results will be presented.

II. SENSOR DESCRIPTION AND MODEL

A. The Sensor

Even if the tracking and pose estimation method presented in this paper can be used with any omnidirectional stereovision system, we used a particular one. It is composed by a unique camera equipped with a telecentric lens and four parabolic mirrors placed in a square at the same distance from the camera, so that their axes are parallel to the camera optical axis (Fig. 1). For more details about the design of this sensor, see [3].

Using parabolic mirrors and a telecentric lens, giving an orthographic projection, allows to keep a central projection for each mirror. So each individual part of the stereo sensor

Guillaume Caron and El Mustapha Mouaddib are with MIS laboratory, University of Picardie Jules Verne, 80000 Amiens, FRANCE; e-mail {guillaume.caron, mouaddib}@u-picardie.fr

Eric Marchand is with INRIA, IRISA, Lagadic, 35000 Rennes, France; e-mail {Eric.Marchand}@irisa.fr

can be modelled using the equivalent sphere projection model.

B. Unified Central Projection Model

Since the work of Barreto *et al.* [10], a unified projection model for central projection cameras was designed. This model describes a family of cameras from perspective to catadioptric ones with particular shape mirrors. The paraboloidal mirror we use is one of them.

According to this model, a central projection camera can be modelled by a first projection on a sphere with coordinates $(0, 0, \xi)$ in the camera frame followed by a perspective projection on the image plane. Such a model can be defined using parameter ξ which depends intrinsically of the catadioptric camera mirror parameters.

Knowing intrinsic parameters $\gamma = \{p_x, p_y, u_0, v_0, \xi\}$, a 3D point $\mathbf{X} = (X, Y, Z)$ is first projected on a unitary sphere and then in the image plane as $\mathbf{x} = (x, y, 1)$. The relationship between $\mathbf{X}$ and $\mathbf{x}$ can be expressed as:

$$\mathbf{x} = pr_\gamma(\mathbf{X}) \quad \text{with} \quad \begin{cases} x = \dfrac{X}{Z + \xi\sqrt{X^2+Y^2+Z^2}} \\ y = \dfrac{Y}{Z + \xi\sqrt{X^2+Y^2+Z^2}} \end{cases} \tag{1}$$

$\mathbf{x}$ is the point on the virtual normalized plane and the image point in pixellic coordinates is obtained by:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} p_x & 0 & u_0 \\ 0 & p_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{2}$$

C. Stereo Model

We chose to model our stereo sensor including one camera and four parabolic mirrors as four cameras, one for each mirror. Each of these cameras handles a set of intrinsic parameters γ_i. To model our rig, we fix the first camera/mirror as the camera frame origin. Hence, the three other cameras are placed relatively to the first one. We note ${}^{c_2}\mathbf{M}_{c_1}$, ${}^{c_3}\mathbf{M}_{c_1}$, ${}^{c_4}\mathbf{M}_{c_1}$ these relative poses which are part of the full calibrated stereo system. One can note this model is expendable with more cameras, knowing each ${}^{c_j}\mathbf{M}_{c_1}$.

III. POSE ESTIMATION APPROACH

A. Overview

The proposed approach is based on works of Marchand *et al.* [7], [8] where the pose computation problem is defined as a virtual visual servoing one. This virtual servoing process is similar to a full scale non-linear estimation method of the pose. This method assumes the stereo system calibration parameters are known.

B. Method

The virtual monocular camera is defined by its projection function $pr_\gamma()$ and its position in the object frame by an homogeneous matrix ${}^c\mathbf{M}_o$. A 3D object have several features ${}^o\mathbf{P}$ defined in its own frame. The method will estimate the real pose by minimizing the error Δ between the observed data, or detected features, $\mathbf{s}^*$ and the position $\mathbf{s}$ of the same features computed by forward-projection according to the current pose. So with k features, we have

$$\Delta = \sum_{i=1}^{k} (pr_\gamma({}^c\mathbf{M}_o, {}^o\mathbf{P}_i) - s_i^*)^2 \tag{3}$$

With this formulation, a virtual camera with inital pose ${}^{c_0}\mathbf{M}_o$, is moved using a visual servoing control law to minimize the error Δ. At convergence, the virtual camera reaches the pose ${}^{c^*}\mathbf{M}_o$ which minimizes Δ. Assuming this non-linear estimation process converges, this pose is the real camera pose.

C. Stereo Extension

As stated before, we want to apply this method to an omnidirectional stereovision sensor considered as a rig of four catadioptric cameras. Dionnet *et al.* in [7], modelled the virtual visual servoing problem for a two perspective cameras stereo rigid system saying that knowing the system calibration (two sets of intrinsic parameters and the second camera pose w.r.t. the first), we can rewrite the Δ criterion to take into account two cameras. It can be generalized to N cameras knowing each of N sets of intrinsic parameters and each $N-1$ relative camera poses w.r.t. the first one. So, we can write

$$\Delta = \sum_{j=1}^{N} \sum_{i=1}^{k_j} (pr_{\gamma_j}({}^{c_j}\mathbf{M}_{c_1}\, {}^{c_1}\mathbf{M}_o, {}^o\mathbf{P}_i) - {}^{c_j}s_i^*)^2 \tag{4}$$

with ${}^{c_1}\mathbf{M}_{c_1} = \mathbf{I}_{4\times4}$. With this formulation, only 6 parameters have to be estimated, as for the monocular pose estimation problem. For instance, if $N = 2$, we retrieve the two cameras case. In our four "cameras" stereo case, $N = 4$. Anyway, assuming that $\mathbf{r}$ is a vector representation of the pose ${}^{c_1}\mathbf{M}_o$, this remains to minimize a residual Δ defined as

$$\Delta = \sum_{i=1}^{k} (s_i(\mathbf{r}) - s_i^*)^2 = ||\mathbf{s}(\mathbf{r}) - \mathbf{s}^*||^2 \tag{5}$$

The error to be regulated is hence $\mathbf{e} = \mathbf{s}(\mathbf{r}) - \mathbf{s}^*$. The primitives motion is linked to the virtual camera velocity by $\dot{\mathbf{e}} = -\lambda\mathbf{e}$, only depending of $\dot{\mathbf{s}}$ which can be written as

$$\dot{\mathbf{s}} = \frac{\mathrm{d}\mathbf{s}}{\mathrm{d}t} = \frac{\partial\mathbf{s}}{\partial\mathbf{r}} \frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t} = \mathbf{L}_\mathbf{s}\mathbf{v}. \tag{6}$$

$\mathbf{L}_s$ is called the interaction matrix and links the feature motion in the image to the camera velocity $\mathbf{v}$.

D. Robust Stereo Pose Estimation

In order to have a precise estimation, $\mathbf{s}^*$ must have a sufficient precision. So, as in [7], we use a M-estimator [9] to deal with outliers. A lot of functions have been proposed in the literature. They allow uncertain measures to be less likely considered and in some cases completely rejected. To add robust estimation to our objective function, it is modified by

$$\Delta_\mathcal{R} = \sum_{i=1}^{k} \rho(s_i(\mathbf{r}) - s_i^*), \tag{7}$$

where $\rho(u)$ is a robust function that grows subquadratically and is monotonically non-decreasing with increasing $|u|$. Iterative Re-Weighted Least Squares is a common method of applying the M-Estimator. Thus, the error to be regulated to 0 is defined, in a matricial form, as:

$$\mathbf{e} = \mathbf{D}(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*), \tag{8}$$

where $\mathbf{D}$ is a diagonal weighting matrix given by $\mathbf{D} = diag(w_1, ..., w_k)$. Each w_i is a weight given to specify the confidence in each feature location. The computation of these weights is described in [11].

A simple control law that allows to move a virtual camera can be designed to try to ensure an exponential decoupled decrease of $\mathbf{e}$. It is given by:

$$\mathbf{v} = -\lambda(\mathbf{D}\mathbf{L}_s)^{+}\mathbf{D}(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*), \tag{9}$$

where $\mathbf{v} = (v, \omega)$ is the virtual camera velocity with v, the instantaneous linear velocity and ω the intantaneous angular camera velocity. The interaction matrix $\mathbf{L}_s$ is defined in equation (6) and λ is a gain that tunes the convergence rate. The computation of interaction matrix will be discussed in section IV.

Considering the minimisation of equation (4), with $N = 4$, since we know the relation between mirrors poses (${}^{c_j}\mathbf{M}_{c_1}$), we can operate a frame change of the pose velocity vector, in order to express it in each mirror frame. For instance, mirror 1 has a velocity vector $\mathbf{v}_1$ and mirror j, a velocity vector $\mathbf{v}_j$. We can express $\mathbf{v}_j$ w.r.t. $\mathbf{v}_1$:

$$\mathbf{v}_j = {}^{c_j}\mathbf{V}_{c_1}\mathbf{v}_1, j = 2..4 \tag{10}$$

where ${}^{c_j}\mathbf{V}_{c_1}$ is the twist transformation matrix:

$${}^{c_j}\mathbf{V}_{c_1} = \begin{bmatrix} {}^{c_j}\mathbf{R}_{c_1} & [{}^{c_j}\mathbf{t}_{c_1}]_\times \\ 0 & {}^{c_j}\mathbf{R}_{c_1} \end{bmatrix}. \tag{11}$$

So the feature velocity in "image" j can be related to the motion of camera 1 by

$$\dot{\mathbf{s}}_j = \mathbf{L}_j\mathbf{v}_j = \mathbf{L}_j{}^{c_j}\mathbf{V}_{c_1}\mathbf{v}_1, \tag{12}$$

and, for four cameras, we have:

$$\begin{bmatrix} \dot{\mathbf{s}}_1 \\ \dot{\mathbf{s}}_2 \\ \dot{\mathbf{s}}_3 \\ \dot{\mathbf{s}}_4 \end{bmatrix} = \begin{bmatrix} \mathbf{L}_1 \\ \mathbf{L}_2{}^{c_2}\mathbf{V}_{c_1} \\ \mathbf{L}_3{}^{c_3}\mathbf{V}_{c_1} \\ \mathbf{L}_4{}^{c_4}\mathbf{V}_{c_1} \end{bmatrix} \mathbf{v}_1. \tag{13}$$

Finally, we get the following control law, with only six parameters to estimate:

$$\mathbf{v}_1 = -\lambda \begin{bmatrix} \mathbf{D}_1\mathbf{L}_1 \\ \mathbf{D}_2\mathbf{L}_2{}^{c_2}\mathbf{V}_{c_1} \\ \mathbf{D}_3\mathbf{L}_3{}^{c_3}\mathbf{V}_{c_1} \\ \mathbf{D}_4\mathbf{L}_4{}^{c_4}\mathbf{V}_{c_1} \end{bmatrix}^{+} \begin{bmatrix} \mathbf{D}_1 \\ \mathbf{D}_2 \\ \mathbf{D}_3 \\ \mathbf{D}_4 \end{bmatrix} \begin{bmatrix} \mathbf{s}_1(\mathbf{r}_1) - \mathbf{s}_1^* \\ \mathbf{s}_2(\mathbf{r}_2) - \mathbf{s}_2^* \\ \mathbf{s}_3(\mathbf{r}_3) - \mathbf{s}_3^* \\ \mathbf{s}_4(\mathbf{r}_4) - \mathbf{s}_4^* \end{bmatrix}. \tag{14}$$

The pose ${}^{c_1}\mathbf{M}_o$ is then updated using the exponential map of $se(3)$ (see [12] for details)

$${}^{c_1}\mathbf{M}_o^{t+1} = {}^{c_1}\mathbf{M}_o^t e^{[\mathbf{v}_1]}, \tag{15}$$

and poses of the three other cameras are then updated using estimated stereo rig calibration parameters ${}^{c_j}\mathbf{M}_{c_1}$: ${}^{c_j}\mathbf{M}_o = {}^{c_j}\mathbf{M}_{c_1}\,{}^{c_1}\mathbf{M}_o$ and will be used in equation (14) to compute $\mathbf{s}_j(\mathbf{r}_j)$ and ${}^{c_j}\mathbf{V}_{c_1}$.

The feature type choice and hence the interaction matrix expression is a key point of this algorithm and is described in section IV.

E. Pose And Calibration Parameters Estimation

In equation (6), the system is supposed to be calibrated but it is possible to relax this knowledge adding the calibration parameters to the estimation process with

$$\dot{\mathbf{s}} = \frac{\mathrm{d}\mathbf{s}}{\mathrm{d}t} = \frac{\partial \mathbf{s}}{\partial \mathbf{r}}\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t} + \frac{\partial \mathbf{s}}{\partial \gamma}\frac{\mathrm{d}\gamma}{\mathrm{d}t}. \tag{16}$$

This time, there are two velocity vectors, still the pose one but the intrinsic parameters velocity vector too, meaning it is possible to vary the intrinsic parameters in the same optimisation process. Following the same idea, it is possible to relax relative poses ${}^{c_j}\mathbf{M}_{c_1}$ between cameras and inserting them in the optimisation process to have a full stereo calibration method. This method is used to calibrate our stereo rig offline using points.

IV. VISUAL FEATURES

A. Feature Type

The tracked object is defined as a 3D model made of 3D lines. So as it has been done in [8] and other papers, we consider as visual features the distance between a point $\mathbf{p}$, detected in the image and the projection of a line, a conic $\mathbf{c}(\mathbf{r})$ in the image plane, for a given pose. The vector $\mathbf{s}(\mathbf{r})$ will therefore be defined by:

$$\mathbf{s}(\mathbf{r}) = \begin{bmatrix} \vdots \\ s_i(\mathbf{r}) \\ \vdots \end{bmatrix} \quad \text{with} \quad s_i(\mathbf{r}) = d_a(\mathbf{x}, \mathbf{c}(\mathbf{r})) \tag{17}$$

where $d_a()$ defines the algebraic distance (detailled in section IV.D.) between point $\mathbf{x}$ and the projection $\mathbf{c}(\mathbf{r})$ of the 3D line.

B. Projection of a 3D Straight Line

As in [8], we modelled a 3D straight line by the intersection of two planes, one including the equivalent sphere center and the other perpendicular to the first. This is not the only possible 3D line representation, Pluker coordinates or a 3D point and a vector are other possible representations.

These two planes $\mathcal{P}_1$, $\mathcal{P}_2$ are defined by:

$$\begin{aligned} \mathcal{P}_1 &: A_1X + B_1Y + C_1(Z - \xi) = 0 \\ \mathcal{P}_2 &: A_2X + B_2Y + C_2Z + D_2 = 0 \end{aligned} \tag{18}$$

with the following constraints on the 3D parameters:

$$\begin{cases} A_1^2 + B_1^2 + C_1^2 = 1 \\ A_2^2 + B_2^2 + C_2^2 = 1 \\ A_1A_2 + B_1B_2 + C_1C_2 = 1 \end{cases} \tag{19}$$

so that the two planes with unit normals $\mathbf{N}_1 = (A_1, B_1, C_1)$ and $\mathbf{N}_2 = (A_2, B_2, C_2)$ are orthogonals.

Following the projection model defined by equation (1), the projection of a straight line in the image is the perspective projection of the circle defined as the intersection between

the plane $\mathcal{P}_1$ and the unitary sphere $\mathcal{S}$ centered in $(0,0,\xi)$. Following the process presented in [8], it is possible to define the conic equation with $\mathcal{P}_1$, $\mathcal{P}_2$ and $\mathcal{S}$ parameters by:

$$Q(x,y) = a_0x^2 + a_1y^2 + 2a_2xy + 2a_3x + 2a_4y + 1 \quad (20)$$

with

$$\begin{cases} a_0 = \frac{A_1^2}{C_1^2}(1-\xi^2) - \xi^2 & a_1 = \frac{B_1^2}{C_1^2}(1-\xi^2) - \xi^2 \\ a_2 = \frac{A_1B_1}{C_1^2}(1-\xi^2) & a_3 = \frac{A_1}{C_1} \\ a_4 = \frac{B_1}{C_1} \end{cases} \quad (21)$$

C. Interaction Matrix for a Line

To compute the interaction matrix, [8] show it is possible to rewrite a_0, a_1 and a_2 using a_3 and a_4 so that their time derivative are functions of $\dot{a}_3$ and $\dot{a}_4$. Hence, determining interaction matrices $\mathbf{L}_{a_3}$ and $\mathbf{L}_{a_4}$ will lead to the three others. Following [13], and defining $\alpha = -\frac{A_2}{D_2} + \frac{C_2}{D_2}a_3$, $\beta = -\frac{B_2}{D_2} + \frac{C_2}{D_2}a_4$, we obtain for a line and one camera:

$$\begin{aligned} \mathbf{L}_{a_4} &= \begin{bmatrix} \beta a_3 & \beta a_4 & \beta & 1+a_4^2 & -a_3a_4 & -a_3 \end{bmatrix} \\ \mathbf{L}_{a_3} &= \begin{bmatrix} \alpha a_3 & \alpha a_4 & \alpha & a_3a_4 & -1-a_3^2 & a_4 \end{bmatrix} \\ \mathbf{L}_{a_2} &= (1-\xi^2)(a_3\mathbf{L}_{a_4} + a_4\mathbf{L}_{a_3}) \\ \mathbf{L}_{a_1} &= 2a_4(1-\xi^2)\mathbf{L}_{a_4} \\ \mathbf{L}_{a_0} &= 2a_3(1-\xi^2)\mathbf{L}_{a_3} \end{aligned} \quad (22)$$

D. Interaction Matrix for Point-Conic Distance Feature

The chosen feature is the distance between a point and the projection of a 3D line, a conic in the used projection system. We decided to use the algebraic distance. Considering a point (x,y), its algebraic distance from a conic is:

$$d_a = Q(x,y) \quad (23)$$

with $Q(x,y)$ defined in equation (20). The interaction matrix $\mathbf{L}_{d_a}$ related to this distance is given by [8], considering the time derivative of d_a,

$$\dot{d}_a = \dot{a}_0x^2 + \dot{a}_1y^2 + 2\dot{a}_2xy + 2\dot{a}_3x + 2\dot{a}_4y \quad (24)$$

we immediately obtain:

$$\mathbf{L}_{d_a} = \begin{bmatrix} x^2 \\ y^2 \\ 2xy \\ 2x \\ 2y \end{bmatrix}^T \begin{bmatrix} \mathbf{L}_{a_0} \\ \mathbf{L}_{a_1} \\ \mathbf{L}_{a_2} \\ \mathbf{L}_{a_3} \\ \mathbf{L}_{a_4} \end{bmatrix} \quad (25)$$

So there will be four $\mathbf{L}_{d_a}$, one for each camera and they will be combined as shown in equation (14).

V. IMAGE PROCESSING

To find corresponding edges, we use the moving edge algorithm [14]. The idea is to sample contours at a regular step and to use an oriented gradient mask to find corresponding contour by convolution maximization along a range search (see [8] for an explanation of this process).

To sample contours, it is possible to sample regularly the 3D straight line and then project these 3D sample points in the image. But it is more efficient to directly sample the conic. Sturm *et al.* [15] expressed a relationship between a conic and a circle by an eigendecomposition of a conic matrix. Conic parameters are known from equation (20). So we can form its matrix representation,

$$\mathbf{C} = \begin{pmatrix} a_0 & a_2 & a_3 \\ a_2 & a_1 & a_4 \\ a_3 & a_4 & 1 \end{pmatrix} \quad (26)$$

and decompose it

$$\mathbf{C} = \mathbf{R}\mathbf{\Sigma}\mathbf{R}^T. \quad (27)$$

Then, using $\mathbf{\Sigma}$ and $\mathbf{R}$, the conic to unitary circle transformation $\mathbf{T}$ is given by

$$\mathbf{T} = \mathbf{\Sigma}^{-1}\mathbf{R}^{-1} \quad (28)$$

which first rotates the conic to fit the standard 2D frame and then normalizes its axes to fit the unitary circle. Starting and ending points of the edge are also known so that the range to sample is known. The inverse transformation $\mathbf{T}^{-1}$ is finally applied to each sample to come back on the conic.

Each sample is the starting point of corresponding contour search. The moving edge algorithm searches this correspondence along the contour normal at each sample. To compute this normal, we can directly use the partial derivates of $Q(x,y)$ as:

$$\begin{pmatrix} \frac{\partial Q}{\partial x} \\ \\ \frac{\partial Q}{\partial y} \end{pmatrix} = \begin{pmatrix} 2a_0x + 2a_2y + 2a_3 \\ \\ 2a_1y + 2a_2x + 2a_4 \end{pmatrix} \quad (29)$$

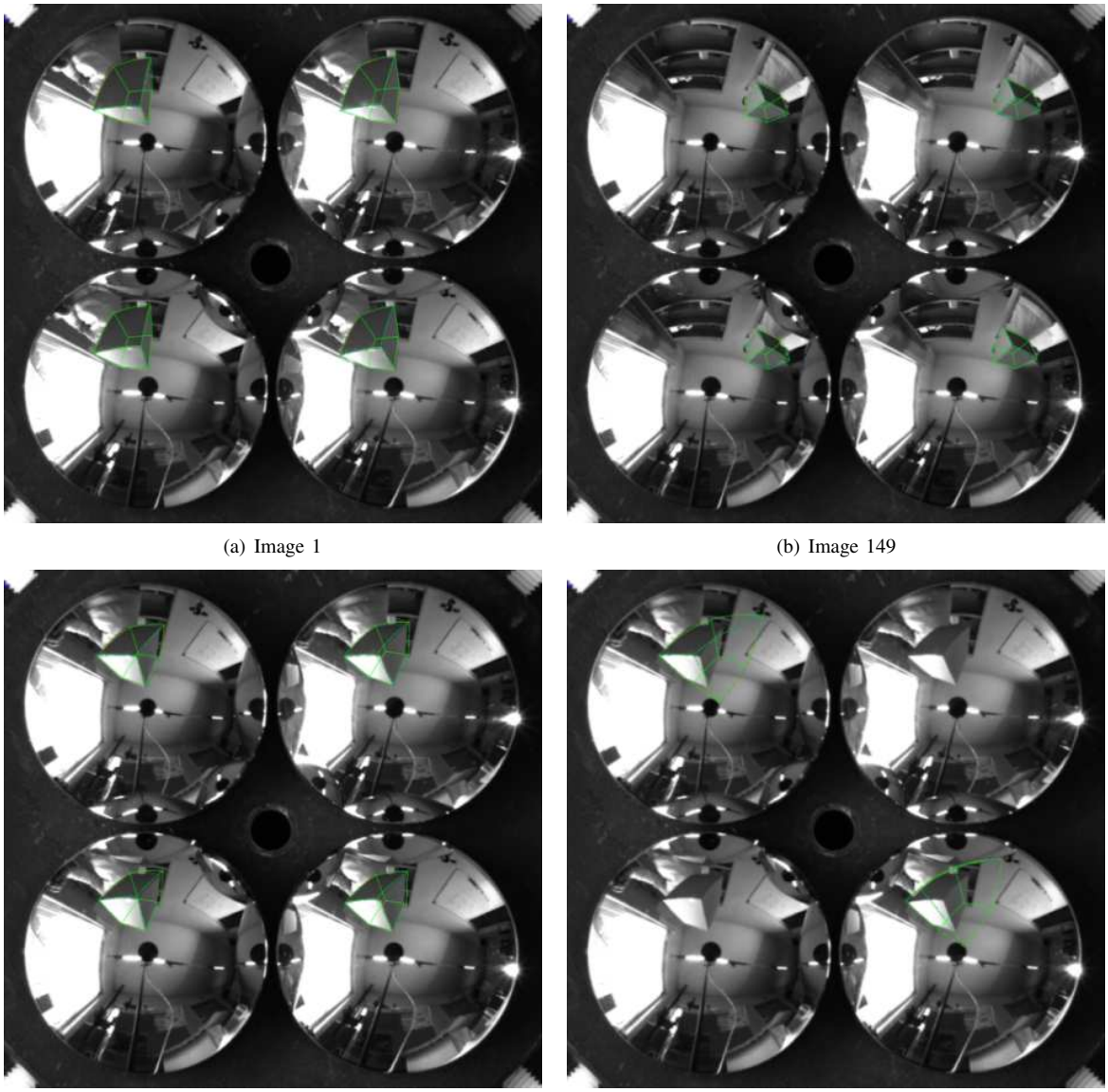
VI. RESULTS

The algorithm was applied on real image sequences. We used the ViSP library [16] to develop the algorithm. Frames are acquired at 10 fps with a resolution of 1280×960 pixels and mean processing time is about 300ms.

A. Tracking a Box

In this experiment, the sensor is immobile and a box ($30cm \times 25cm \times 20cm$) is handheld and moves in a large part of the sensor field of view (Fig. 2), at a distance from the sensor between 35cm and 75cm. Despite disturbing image gradients in the background and illumination variations, the tracking is achieved all along the sequence of 312 images. Of course, some sample points are influenced by the strong background gradients but thanks to robust estimation and robustness induced by stereovison, these ones are rejected.

We also applied tracking and pose estimation with our method to only two mirrors of the sensor, the ones with the widest baseline (the top-left and the bottom-right mirrors). These two mirrors should be the ones used in a more classical stereo sensor: a two mirrors/cameras rig. But with two mirrors the object is lost from image 277 (Fig. 2(d)). This shows the higher robustness, due to information redundancy, brought by the two additional mirrors. The video of this experiment is available from the research section in website `http://mis.u-picardie.fr/~g-caron/en/`.



(a) Image 1

(b) Image 149

(c) Image 312

(d) Image 312 with two mirrors pose estimation. The object is lost from image 277. To be compared with Fig. 2(c)

Fig. 2. Images of pose estimation for the handheld box sequence.

B. Auto-occlusion

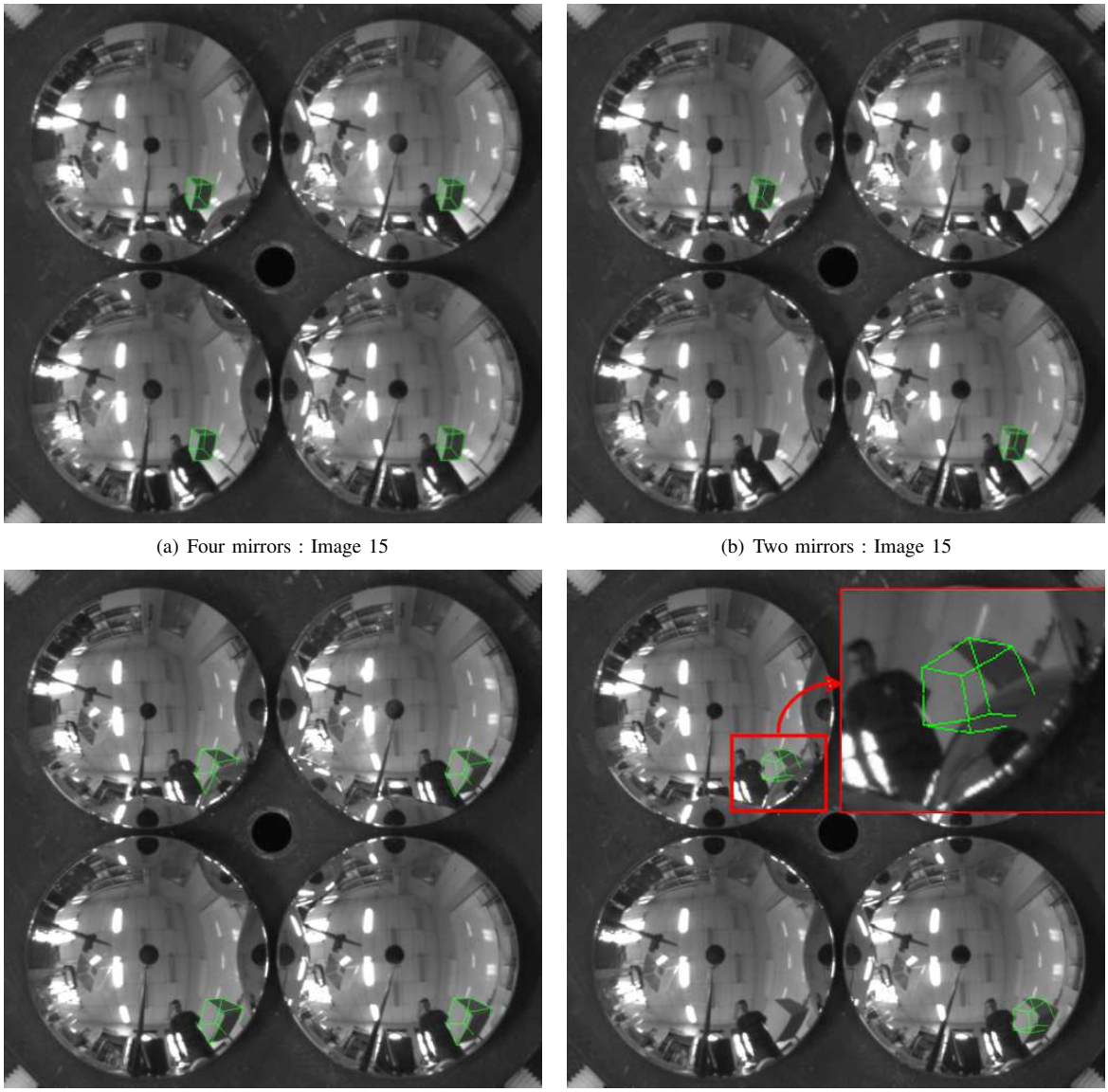
Another experiment has been made in order to show the higher robustness induced by the four mirrors stereo sensor w.r.t. a two mirrors stereo sensor. A problem of auto-occlusion can appear when placing parabolic mirrors on the same plane. They reflect on each other. This phenomenon creates problematic image zones and these are generally withdrawn thanks to an image mask. We also used this method to avoid tracking problems and we made a second experiment in which the box starts outside the inter-reflection zone between the two diagonal mirrors, moving through it and come back near starting position. The four mirrors stereo pose estimation and tracking process succeed (Fig. 3(a) and 3(c)) while with two mirrors, the box is lost in the auto-occlusion zone (Fig. 3(b) and 3(d)).

VII. CONCLUSION

We have presented a robust 3D model-based pose estimation method using omnidirectional stereovision. Results with our four mirrors sensor show even higher robustness than a two mirrors approach. The combination of VVS and four mirrors omni-stereo sensor has proven to be robust to disturbing background or even to auto-occlusion thanks to redundancy, all around the sensor. Future works will be focused on the full utilization of the stereo sensor relaxing the constraint of knowing the 3D model.

REFERENCES

[1] G. CARON, E. M. MOUADDIB, "Vertical Line Matching for Omni-directional Stereovision Images", *IEEE Int. Conf. on Robotics and Automation*, Kobe, Japan, may 2009.



(a) Four mirrors : Image 15

(b) Two mirrors : Image 15

(c) Four mirrors : Image 153

(d) Two mirrors : Image 153

Fig. 3. Images of pose estimation in presence of auto-occlusion. The robustness induced by the four mirrors rig allows to deal with auto-occlusions.

[2] E. MOUADDIB, R. SAGAWA, T. ECHIGO AND Y. YAGI, "Stereo Vision with a Single Camera and Multiple Mirrors", *IEEE Int. Conf. on Robotics and Automation*, Barcelona, Spain, april 2005.

[3] G. DEQUEN, L. DEVENDEVILLE, E. MOUADDIB, "Stochastic Local Search for Omnidirectional Catadioptric Stereovision Design", *Iberian Conference on Pattern Recognition and Image Analysis*, Estoril, Portugal, may 2007.

[4] V. SUNDARESWARAN, R. BEHRINGER, "Visual servoing-based augmented reality", *IEEE Int. Workshop on Augmented Reality*, San Francisco, USA, nov 1998.

[5] E. MARCHAND, F. CHAUMETTE, "Virtual visual servoing: a framework for real-time augmented reality", *EUROGRAPHICS*, Saarbrücken, Germany, sept 2002.

[6] A.I. COMPORT, E. MARCHAND, F. CHAUMETTE, "Robust model-based tracking for robot vision", *IEEE Int. Conf. on Inteligent Robots and Systems*, Sendai, Japan, sept 2004.

[7] F. DIONNET, E. MARCHAND, "Robust Stereo Tracking for Space Applications", *IEEE Int. Conf. on Inteligent Robots and Systems*, San Diego, USA, nov 2007.

[8] E. MARCHAND, F. CHAUMETTE, "Fitting 3D Models on Central Catadioptric Images", *IEEE Int. Conf. on Robotics and Automation*, Roma, Italy, april 2007.

[9] P.-J. HUBER, *Robust Statistics*, Wiler, New York, USA, 1981.

[10] J.P. BARRETO, H. ARAUJO, "Issues on the Geometry of Central Catadioptric Images", *Int. Conf. on Computer Vision and Pattern Recognition*, Hawai, USA, dec 2001.

[11] A.I. COMPORT, E. MARCHAND, M. PRESSIGOUT, F. CHAUMETTE, "Real-time markerless tracking for augmented reality: the virtual visual servoing framework", *IEEE Trans. on Visualization and Computer Graphics*, july/aug 2006.

[12] Y. MA, S. SOATTO, J. KOŠECKÁ, S. SASTRY, "An invitation to 3-D vision", *Springer*, 2004.

[13] B. ESPIAU, P. RIVES, "Closed-loop recursive estimation of 3d features for a mobile vision system", *IEEE Int. Conf. on Robotics and Automation*, Raleigh, USA, mar 1987.

[14] P. BOUTHEMY, "A maximum likelihood framework for determining moving edges", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, may 1989.

[15] P. STURM, P. GARGALLO, "Conic Fitting Using the Geometric Distance", *Asian Conference on Computer Vision*, Tokyo, Japan, nov 2007.

[16] E. MARCHAND, F. SPINDLER, F. CHAUMETTE, "ViSP for visual servoing: a generic software platform with a wide class of robot control skills", *IEEE Robotics and Automation Magazine*, dec 2005.